AD-A246 744

(1)

DTIC
ELECTE
MAR 03 1992
S D

EDUCATION NECESSARY FOR AIR FORCE
SOFTWARE MANAGERS TO USE THE ADA PROGRAMMING
LANGUAGE AND SOFTWARE ENGINEERING EFFECTIVELY

THESIS

James W. Worley, Jr.
Captain, USAF

AFIT/GSS/ENG/91D-12

92-04869

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

92 2 25 132

AFIT/GSS/ENG/91D-12

EDUCATION NECESSARY FOR AIR FORCE SOFTWARE MANAGERS TO USE THE ADA PROGRAMMING LANGUAGE AND SOFTWARE ENGINEERING EFFECTIVELY

THESIS

James W. Worley, Jr.
Captain, USAF

AFIT/GSS/ENG/91D-12

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information is contained therein. Furthermore, the views expressed in the document are those of the author and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.

AFIT/GSS/ENG/91D-12

EDUCATION NECESSARY FOR AIR FORCE SOFTWARE MANAGERS TO USE THE ADA PROGRAMMING LANGUAGE AND SOFTWARE ENGINEERING EFFECTIVELY

THESIS

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Software Systems Management

James W. Worley, Jr.

Captain, USAF

December 1991

Approved for public release; distribution unlimited

Acknowledgements

I would like to express my sincere appreciation to Captain Dawn Guido for advising and guiding this thesis effort toward a result that could benefit many people throughout the Air Force. I especially appreciate her trusting that I would finish the work with minimum external motivation.

I would like to thank Lieutenant Colonel Pat Lawless for giving her blessing to pursue the research and for her guidance and feedback throughout the effort. Also, I extend my thanks to Major Tom Croak for believing that this research was important enough to sponsor and for providing insight into the problems software personnel face.

A special thanks to AFIT Professor, Dr Guy Shane for lending his statistical and research expertise to make this thesis a valid product.

Finally, during what has been an illustrious military career, I have set many goals and, with God's help, I have accomplished them all. Each goal has required a different level of personal sacrifice but perhaps none so much as completing this thesis (and in parallel the graduate program). This sacrifice has been the time I lost with my immediate family because I was absorbed in school work. Though the rewards of this effort may be plenty, nothing can get back that time. It is to these ends that I dedicate this work to God and the three people who have sacrificed with me: my wife Patty, my daughter Dusti, and my son James III.

Table of Contents

Page

List of Tables

AFIT/GSS/ENG/91D-12

Abstract

This research investigated the perceived Ada and software engineering education needs of software managers throughout the Air Force and provided recommendations to Air Staff for developing the education. Since the Department of Defense mandated all DOD agencies use the Ada programming language for software development, the education of the personnel who have to support the policy has lagged behind the implementation. Ada is not a simple language to use and is only fully effective if used with sound software engineering principles.

A survey was given to Air Force mid-level personnel who manage software in some capacity. Of the software personnel surveyed, 48 percent indicated that Ada education would enhance their ability to perform their jobs and 83 percent indicated software engineering education would help. Less than 45 percent of the personnel who work with Ada have had a formal course in using the language while over 70 percent of the people have had some form of formal software engineering education.

This research also investigated the required frequency of education. The respondents indicated that the frequency (periodic or as-needed) should depend on the nature of the individuals job and rank or position.

EDUCATION NECESSARY FOR AIR FORCE

SOFTWARE MANAGERS TO USE THE ADA PROGRAMMING

LANGUAGE AND SOFTWARE ENGINEERING EFFECTIVELY

I. Introduction

General Issue

To help reduce runaway software costs, the DOD sponsored the development of the Ada programming language. Current DOD policy specifies all DOD agencies use Ada to develop their software systems. Few Air Force software managers know how to use the Ada programming language, in concert with software engineering, to develop software that inherently offers lifecycle cost savings and benefits. An Air Force Broad Area Review (BAR) on Software Management states that improved software engineering education within the Air Force will help solve this problem (Air Force, 1989:19).

Management Question

Do software managers need more education about software engineering principles (which are key to lifecycle benefits) and Ada's adherence to these principles? If so, what education will enable them to apply software engineering principles effectively and use Ada to develop, support, or manage the software process?

Investigative Questions

This issue, originating at the DOD level, is of special interest to the Air Force Headquarters Software Management office (HQAF/SCXS).

Therefore, HQAF/SCXS sponsored this research. To address Air Staffs' concerns about the topic, software managers at various Air Force functional levels answered the following questions.

1. What formal software education have they received?

2. Are they knowledgeable about the software engineering principles Ada supports?

3. Are they aware of the advantages of using Ada as a software engineering tool?

4. Do they believe additional knowledge about the Ada language and software engineering would improve their ability to perform their job?

5. What formal education would enhance their ability to perform their job?

6. How often should the formal education occur?

7. Who should fund the education?

Scope of Research

This thesis does not advocate that the Ada language alone is a cure-all to programming language problems. It does recognize however, that Ada and software engineering used together offer many benefits. These benefits include reducing risk and decreasing software development cost and easing the burden and cost of maintaining the software throughout its lifecycle (Deputy, 1990).

The scope of this research is to define what education Air Force software managers need to effectively combine Ada with software engineering principles to develop, support, and manage the software process. It also investigates when and how to provide the education.

This research does not investigate software engineering or the Ada programming language as lone entities. It does not discuss programming languages other than Ada or software development methods that do not employ software engineering principles (such as unstructured design). This research also excludes informal education methods such as self-study and work experience.

This research reviews current software education practices in both the Department of Defense (DOD) and commercial sectors. Their potentially varied perspectives on the problem provided valuable insight to the management questions.

Background

In the 1970s, DOD sponsored an effort to develop a programming language (eventually named Ada) to cope with increasing software complexity and size. The language was not to be just another 'programming' language. Instead, it was specifically "... designed to support software engineering principles." (Booch, 1983:xv).

Subsequent to the development of Ada, DOD set policy that all DOD agencies use Ada to develop all DOD embedded software systems. This policy has evolved from mandating the use of Ada just for embedded software systems to using Ada to develop and maintain all DOD software systems (Department of Defense, 1987a and 1987b).

Unfortunately, most managers did not know how to use Ada effectively when this policy first came about. A study sponsored by the Ada Joint Program Office (AJPO) reported, "Software engineering training and management awareness of training in Ada are currently inadequate." (AJPO, 1987:III-1). This research defines the formal education needs of software managers who work with Ada. It also recommends when and how to provide the education.

Thesis Outline

Above is an introduction to the problem this thesis addresses with a brief background on why the problem exists. Chapter 2 of this thesis gives a broader discussion of the background. Chapter 3 describes the methodology to perform the research. Chapter 4 contains the results of the data analysis. Chapter 5 discusses conclusions and recommendations. Various appendices contain the survey and survey information, statistical analyses results, statistical programs, and the raw data. Also included is a Glossary of Terms (Appendix A) to familiarize the reader with software engineering terminology.

II. Background

In the 1970s, the Department of Defense realized it needed a way to reduce the runaway costs of its software development efforts. In 1973 software development made up about 46 percent of all DOD computer costs (Booch, 1983:14). To deal with this 'software crisis', the DOD sponsored the development of the Ada programming language. The millions of dollars in 1973 grew to $20 billion in 1988 and more than $30 billion in 1990. In fact, the amount of DOD software code on order in 1990 exceeded the amount of code that currently existed (Marsh, 1990:62). After Ada was created, the DOD set policy for all DOD agencies to use Ada to develop software for embedded computer systems. Since Ada's birth, DOD's policy has evolved into using Ada to develop and maintain all DOD systems software (DOD, 1987).

However, a study sponsored by the Ada Joint Program Office reported a problem in supporting the policy. The study, performed by an Ada Software Engineering Education and Training (ASEET) team, reported, "Software engineering training and management awareness of training in Ada are currently inadequate." (AJPO, 1987:III-1). To lessen the problem, the DOD should educate its software managers on development methodologies that employ software engineering principles and, in turn, effectively employs the Ada programming language (Croak, 1990).

The following discusses software engineering attributes, the software lifecycle, the inherent characteristics of using Ada in the context of software engineering, what approaches some educators and institutions take to teach Ada, and what academic environments exist to provide the education.

Software Engineering Attributes

Software engineering (first introduced formally in the late 1960s) has become the standard for current software development. Universities teach it and companies use it - replacing the old 'code and go' software development method. In his textbook titled *Software Engineering*, Ian Sommerville explains that well engineered software is maintainable, reliable, and efficient. The lifecycle costs of a software system with these attributes will be lower than those of a software system without them (Sommerville, 1989:4). These attributes are defined as follows:

> ***Maintainable Software***. Software "... written and documented in such a way that changes can be made without undue costs." (Sommerville, 1989:4).
>
> ***Reliable Software***. Software that can "... perform a required function under stated conditions for a stated period of time." (McManus and Schulmeyer, 1987:13).
>
> ***Efficient Software***. Software that uses system resources such as memory and processing time efficiently (Sommerville, 1989:4).

Software Lifecycle

The software lifecycle is a series of activities. It begins with the recognition of a need and ends with the destruction or retirement of the software (Davis, 1990:379). Lifecycle activities are: requirements analysis, design, development, testing, deployment, and maintenance and evolution support (Davis, 1990:8,379). Managers can apply software engineering principles to each lifecycle activity to reduce the lifecycle costs of the software.

Growing Ada Requirements (U.S., 1990:1-3)

Ada use, in all sectors, grew from about one-seventh of the software market in 1988 to about one-third of the software market in 1990. At this growth rate, experts expect that Ada will be used in at least one-half of all software products by 1995.

DOD Ada Market Influence. The DOD makes up about 80 percent of the Ada product market. This large market share likely exists because DOD developed Ada and has mandated its use in all current and future DOD software system developments. The DOD expects this standardization to curb software development and lifecycle costs. With software development costs for DOD weapon systems alone in the tens of billions of dollars per year, the significance of employing cost-saving software development techniques is clear.

Special Requirements For Ada Education

Grady Booch explains that Ada "... represents a very powerful tool to help us understand problems and express their solutions in a manner that directly reflects the multidimensional real world." (Booch, 1983:xv,3). Traditional programming languages such as FORTRAN, COBOL, C, and Pascal were developed for specific types of applications (for example, FORTRAN for scientific applications and COBOL for business applications). Because of their early development in the history of programming languages, these languages do not reflect more modern design methodologies (Booch, 1983:xv,3). Ada incorporates many features of these languages but adds the dimension of being a vehicle for software engineering (Tomayko, 1989:281). These factors combine to make Ada more complex than more traditional programming languages.

Ada's Complexity. A competent Ada programmer must have a working knowledge of the modularity, abstraction, information hiding, and localization principles of software engineering (Howatt and Umphress, 1989:6). In traditional programming language courses, instructors emphasize learning language syntax. This approach to teaching Ada makes the language appear complex, likely causing the student to produce poorly constructed programs (Booch, 1983:xiv). The intent of the Ada language is to encourage the use of software engineering principles (Howatt and Umphress, 1989:5). Teach it as an engineering tool and not just a programming language (Tomayko, 1989:283).

When to Teach Ada. Ada is useful for a wide variety of application domains. As a result, educators are not sure when to teach it. Dr. Tomayko explains that teaching Ada presents both an opportunity and a dilemma to educators. Because of Ada's inherent ability to support software engineering, the opportunity is available to introduce software engineering at an early stage in a computer science curriculum. The dilemma is that a student's first computer language course in college is usually their first exposure to computer languages. (Tomayko, 1989:281)

Learning a computer language is like learning any foreign language. One must first learn the basics of the language and then build to fluency. Some educators believe the same must occur in teaching computer science. Because Ada presents so much detail outside the basic concept of structured programming, Tomayko believes educators should not teach it until later in the curriculum (Tomayko, 1989:281). Other educators however, feel that Ada is an effective first language *when used properly* (Lawlis, 1991). In essence, there is no general agreement about when to teach Ada.

How to Teach Ada. Dr. Tomayko explains that educators are not sure whether to use Ada as a tool to teach software engineering or to use software engineering as a tool to teach Ada. He has taught both methods and concludes, "... Ada and software engineering are inseparable. Ada should be taught in the context of software engineering ..." (Tomayko, 1989:283).

Current Ada Education Efforts

Air Force Institute of Technology (AFIT) Courses. AFIT offers courses that include the use of Ada through both its professional continuing education courses, which are a part of the Software Professional Development Program (SPDP), and its graduate school.

Graduate Level Education. AFIT does not offer Ada courses per se. Instead, AFIT professors and instructors use Ada in their courses to express concepts such as software engineering, concurrent programming, and environments (Bralick and Umphress, 1988:187). AFIT has three graduate programs designed to educate students about all activities of the software lifecycle. Two programs emphasize the technical aspects of the software lifecycle while the other program emphasizes the managerial aspects.

SPDP Program. As a result of the BAR report, AFIT established a three-year pilot program to update the knowledge of Air Force software professionals (primarily Air Force captains, majors, and their civilian equivalents). The program teaches virtually the same topics as AFIT's graduate courses, but at a much accelerated pace (two-week short courses). Five courses cover the software lifecycle. The course titles are: 1) *Software Engineering Concepts*, 2) *Specification of Software Systems*, 3) *Principles and Applications of Software Design*, 4) *Software Generation and Maintenance*, and 5) *Software Verification and Validation*. (Guido, 1990). In these courses, Ada is used both to express software engineering principles and as an implementation language for project work (Lawlis, 1991).

Both the graduate courses and the SPDP courses provide a much-needed educational base for future DOD software managers.

Air Force Academy. Air Force Academy undergraduate courses teach Ada in the context of software engineering (Anderson and McDonald, 1988:18). This is a move toward providing the Air Force with Ada literate entry-level managers.

Air Force Training School, Keesler Air Force Base, Mississippi. This school offers courses in software engineering and Ada programming. The intended audience is project managers, system configuration managers, designers, senior executives, programmers, and software engineers (Anderson and McDonald, 1988:18).

Civilian Academic Institutions. More than 175 academic institutions offer a total of 265 courses in the Ada programming language, software engineering principles, or both (Anderson and McDonald, 1988:18).

Summary

By DOD directive, using Ada for software systems development is a reality software managers must deal with. The user must be knowledgeable about a tool's characteristics to use it correctly and to its fullest potential. Ada and software engineering are no different. Unfortunately, few Air Force software managers know how to use Ada and software engineering effectively. Yet, they are asked to provide software products that fulfill the expectations of the tools' attributes.

The software community is giving greater attention to the education shortfall. Companies are creating in-house education programs and universities are incorporating software engineering and Ada courses in their curricula. To enhance the knowledge of Air Force personnel who work with software, several Air Force agencies teach a variety of software courses. These courses cover the topic spectrum from software specification through software maintenance and evolution.

Granted, software managers should be knowledgeable about these topics, but not all require the same depth and breadth of knowledge. Managers may need a different level of education because of jobs or positions characteristics. This thesis researches how the Air Force should tailor the education for these software managers, and who should fund it.

III. Methodology

This chapter addresses the research design, the survey description, the respondent description, and the statistical analyses used to accomplish the research.

Research Design

To assure the research process went as smoothly as possible, considerable thought went into the research design. Following is an outline of the design.

Data Collection and Processing. A survey is the data collection instrument. Respondents record their answers to the survey questions on an AFIT Form 11D, *AFIT Data Collection Form.* The forms are then processed on AFIT's optical reader and the data is stored on one of AFIT's mainframe computer systems.

Research Steps. Following are the steps used by the researcher to accomplish the research.

1. Conducted pre-survey discussions with software experts to help focus survey questions toward the problem.
2. Constructed a survey using recommendations from experts and guidelines from literature.
3. Established a plan to process the data.
4. Obtained a letter of endorsement from the sponsor.
5. Obtained approval to distribute the survey from the Air Force Manpower Personnel Center (AFMPC).
6. Contacted distribution focal points.
7. Mailed survey packages to focal points.
8. Statistically analyzed survey responses.
9. Recommend education to enable software managers to more effectively use Ada and software engineering.

Survey Justification

The research investigates Air Force software managers' perceived educational needs. The sample size (68) required for 90 percent statistical confidence (derived with Eq (1)) makes it impractical to contact each person by telephone or personal interview.

Survey Validation

The first step toward developing a valid survey was to ask software experts about what topics they believed were pertinent to this research. The experts were the thesis sponsor, the thesis advisor, and other software educators. Their recommendations were used to develop questions; they reviewed the questions; and the questions were refined as appropriate.

Next, the survey was given to and discussed with students of an AFIT SPDP class. This pre-assessment served two purposes. First, feedback about question construction and survey organization was used to further refine the survey. Second, because the students were formally educated on software engineering before they were surveyed, their responses could be included in the treatment group data-base to study the effects of AFIT's SPDP.

Survey Question Description

The data collection instrument, a 51 question survey (Appendix B), measures certain characteristics about the respondent. Although there are 51 questions, each one falls under one of four categories. These categories are: 1) respondent characteristics and current job(s), 2) background education/training, 3) DOD policy on the Ada programming language, and 4) Ada and software engineering. Following is a description of the categories:

Respondent Characteristics and Current Job. This area questions respondents about their military or civil-service grades, the Air Force commands to which they are assigned, their Air Force specialty codes (AFSC) and job titles, and how long they have been in their current jobs. This area also investigates what percentage of the respondent's current job involves

1. software program management
2. software technical management
3. software development
4. software acquisition
5. software maintenance/testing
6. and other software functions defined by the respondent.

These questions identify the level of management at which the individuals work.

Respondent's Education/Training Background. This section asks respondents what formal software education they have received. In particular, these questions ask how many years ago (if at all) the respondent received formal education/training in computer programming, software engineering, software requirements/specifications, software maintenance/testing, and software design. It also asks how long ago (if at all) the respondent received formal education/training in the Ada, Jovial, COBOL, Fortran, Pascal, Algol, Modula-2, and C programming languages.

DOD Policy on Ada. This section asks the respondents how often they believe managers should be given general and domain-specific education to support this policy. It asks who should provide funds for the education and to what level (depth) managers at different grades should be educated.

Ada and Software Engineering. These questions ask respondents about general Ada characteristics and how Ada relates to software engineering. The section also asks respondents about their perception of what contributions formal education would make to their current jobs.

To facilitate programming the statistical analysis software, each survey question was assigned a name (or abbreviation) that represents the question topic. For instance, 'EADA' would indicate the question that measures the maturity of a respondent's Ada education. See Appendix C for a description of these names.

Variable Assignment

Before analyzing data, independent and dependent variables must be identified. This was accomplished by first identifying survey items that have fixed (independent) response values (such as grade and AFSC). Other factors do not influence their value. Rather, they usually influence the values of other (dependent) factors. These variables were determined by observation (not statistically).

The remaining questions have dependent responses. The responses to these questions may vary according to the independent values determined above.

After identifying the questions, a reliability test was performed to determine the possible grouping of questions into dependent and independent variables. This was accomplished by performing a confirmatory factor analysis on the questions, dependent and independent questions separately, and checking the loading pattern under (software) assigned factors. The results showed that the some of the questions had potential to be grouped into one variable rather than be addressed individually.

Next, questions were grouped into new variables according to the factor analysis discussed above. A Chronbach's alpha analysis was performed on each new variable (of grouped questions) to measured its reliability to perform the intended measurement. New variables that showed a high reliability (greater than 0.5 in this case) were used to measure unitary aspects of the problem.

Here, the variable ADA_JOBS includes the survey questions that ask about the respondent's involvement with Ada. SE_AREAS deals with the respondent's education in the requirements/specification, maintenance/testing, and design *functions of software engineering*. ED_1_NO addresses the need for no education or one time only education. ED_PER addresses the need for periodic education. ADA_ATT deals with attributes of the Ada language. ADA_SE addresses the ability of Ada to support specific software engineering principles. ABILITY addresses education that will help the respondents perform their current job(s). The remaining questions are variables unto themselves (meaning they were not grouped).

Note that the variable assignment analysis was performed before all responses were collected. However, there were sufficient responses (122) to meet the sample requirement of 68 for the required 90 percent statistical confidence interval.

Table 1 shows the assignment of multiple survey questions to single independent variables and the result of the reliability analysis.

TABLE 1

ASSIGNMENT OF QUESTIONS TO INDEPENDENT VARIABLES

Independent Variable (IV) Name	*Question Number Assigned To IV*	*Reliability Results (alpha)*
SE_AREAS	13, 14, 15	0.779395
ADA_JOBS	34, 35, 36, 37, 38	0.879728

Table 2 shows the assignment of multiple survey questions to single dependent variables and the result of the reliability analysis (122 responses).

TABLE 2

ASSIGNMENT OF QUESTIONS TO DEPENDENT VARIABLES

Dependent Variable (DV) Name	*Question Number Assigned To DV*	*Reliability Results (alpha)*
ED_1_NO	24, 26, 27, 30	0.625265
ED_PER	25, 29	0.661957
ADA_ATT	40, 41	0.572352
ADA_SE	42, 43, 44, 45, 46, 47, 48	0.865260
ABILITY	49, 50, 51	0.693829

Respondent Description

This study focuses on those Air Force personnel who manage software systems developed with the Ada programming language. These people include mostly captains, majors, and lieutenant colonels and GS-12s, GS-13s, GS-14s, and GM-14s.

The non-stratified research sample does not address individual characteristics of the population (that is, the needs of GS-12s or Majors in particular). The attribute breakdown only allows the reader to characterize respondents. Whether or not inferences can be made about the population, based on this information, depends on desired statistical confidence.

From information about AFIT SPDP students, the following AFSCs and job specialties were found to involve software related functions.

a. 26XX, Research Scientist (Military).

b. 27XX, Acquisition Officer/Manager (Military).

c. 28XX, Engineer (Military).

d. 49XX, Communications - Computer Systems (Military).

e. 334, Computer Analyst (Civilian).

f. 855, Electronics Engineer (Civilian).

g. 856, Electronics Technician (Civilian).

Respondents with other grades, AFSCs and job specialties are accepted if their jobs involve managing Ada software.

The total estimated population of the above specialty codes (for the grades mentioned) is 7,835 (5,819 civilians and 2,016 military) (Air Force, 89:not numbered). This population includes various software disciplines, some involve the Ada programming language. Exactly how many of these people deal with Ada is uncertain. It is certain however, that software managers involved with Ada are included in the 7,835. Therefore, a population size of 7,835 is used to determine a sample size that guarantees at least a 90 percent confidence interval for the results.

The following equation was used to compute the sample size required for this research. The result (68, rounded up from 67.65) agrees with AFMPC's (equation unknown) computed sample size. AFMPC authorized distribution of 200 surveys which includes an allowance for typical levels of non-response and incomplete responses.

The sample size *n* is defined as,

$$n = \frac{4 * (z^2) * p * (1 - p)}{L^2} \tag{1}$$

where

z = factor of assurance for 90 percent confidence interval (1.645).

p = point estimator that an individual has a particular characteristic (.5 to maximize p*(1-p)).

L = length of confidence interval (.2 for 90%).

(Devore, 1987:255,262,264)

Respondent Selection. Every effort was taken to assure all survey recipients fit the population criteria. To do this, a single point-of-contact was established in several Air Force organizations. These personnel requested and received a specified number of surveys. In turn, they distributed surveys to personnel within their organization who fit the population criteria. Although this process was effective, it makes the sample non-random and could possibly introduce a bias into the data.

Experiment Design For Secondary Research Objective

A quasi-experimental design called a Static-Group Comparison Design was used to accomplish this objective (Emory, 1985:120). This design involves two groups of survey respondents. One group (the 'treatment' group) had nearly or just completed a two week AFIT SPDP course that teaches software engineering principles. The second group (the 'control' group) includes only respondents who participated in the field survey.

The objective of this research (not an investigative question) was to investigate differences between treatment group responses and control group responses to give some indication of the effectiveness of the SPDP courses.

Statistical Analysis

Following is a discussion of statistical tests performed on research data. SAS Institute's *SAS System* software, resident on AFIT's VAX mainframe computer, was used to compute the statistics. Appendix D contains the SAS programs used in this effort.

Frequency Analysis and Descriptive Statistics. Frequency analysis and descriptive statistics are computed for the 51 survey questions for both groups. Appendix E contains tables representing the response frequencies and the descriptive statistics.

Two-Sample t-Test. The two-sample t-test tests the null hypothesis: the means of treatment group responses (to survey questions) are equal to the means of control group responses. This test supports the secondary research objective. If significant differences exist, inferences may be drawn about the impact of the SPDP classes on treatment group responses.

Regression Analysis. Regression analysis tests the linear relationship of performance predictors (independent variables) to performance criteria (dependent variables). This analysis provides a model of the software managers' educational needs.

A stepwise type regression analysis was performed for this research. The stepwise regression adds one predictor at a time and measures its significance to the model. The test results in identifying only those performance predictors significant to the model.

IV. *Analyses and Findings*

This chapter focuses on the collected data and analyses of the data as necessary to answer the investigative questions proposed in Chapter 1. The first section discusses to whom the surveys were distributed and how they responded. The second section provides an analysis of responses to the survey questions as they apply to the investigative questions. This analysis is based on statistical inferences and distribution of responses. The last section discusses the results of the regression analysis performed on the data. Refer to Chapter 3 for a more in-depth discussion of the statistical tests performed.

Survey Distribution and Response Rates

The following demographics show the survey distribution for the control group and the treatment group.

Control Group Distribution and Response Rates. A total 162 surveys were mailed to agencies from seven Air Force (AF) commands and one separate operating agency (SOA). The commands represented were: Communications Command (AFCC), System Command (AFSC), Logistics Command (AFLC), Tactical Air Command (TAC), Military Airlift Command (MAC), Strategic Air Command (SAC), and Air Force Space Command (AFSPACECOM). The SOA was the Air Force Operational Test and Evaluation Center (AFOTEC). Surveys were also given to 15 individuals attending a SPDP course at AFIT and 2 AFIT Software Systems Management graduate students. These SPDP students are unlike the treatment group SPDP students in that they had not yet begun their class.

The total survey distribution was 179. Table 3 shows to whom the surveys were distributed and the number of responses received. Here, 'Others' in column 4 of Table 3 indicates the 17 surveys given to AFIT SPDP and graduate students.

TABLE 3

CONTROL GROUP SURVEY DISTRIBUTION AND RESPONSES BY COMMAND

Command	*Sent to Field*	*Received from Field*	*Received from Others*	*Total Received*
AFCC	16	14	3	17
AFSC	40	18	5	23
AFLC	56	38	5	43
TAC	12	8	0	8
MAC	15	13	3	16
SAC	14	10	1	11
AFSPACECOM	7	4	0	4
AFOTEC	2	2	0	2
Total	*162*	*107*	*17*	*124*

Treatment Group Distribution and Response Rates. A total 70 surveys were given to AFIT students (68 SPDP and 2 graduate). This student body represents a broad number of Air Force commands. Because these surveys were under the direction of AFIT they do not subtract from the 200 surveys allowed by AFMPC. Table 4 shows the survey demographics for these responses. The 'UNKNOWN' category in Tables 4 depicts respondents who did not specify a command.

TABLE 4

COMMAND ASSIGNMENT OF THE TREATMENT GROUP

Command	*Number of Responses*	*Command*	*Number of Responses*
AFCC	12	AFSPACECOM	1
AFSC	19	HQAF	2
AFLC	21	USAFE	1
TAC	4	PACAF	1
MAC	2	ATC	2
SAC	4	UNKNOWN	1

Demographics of Respondents Who Work With Ada. Because some statistical inferences will come from responses of only those people who work with Ada, it is pertinent here to offer the demographics of those people. Table 5 reflects the commands to which respondents who work with Ada are assigned. These respondents are from both the control and treatment groups.

Statistical Analysis of Survey Data Applied to Investigative Questions

The analyses discussed in Chapter 3 were performed to answer the investigative question proposed in Chapter 1. Following is the result of this analysis.

T-test Results. The t-test results show that there is a significant difference between the treatment group and control group means of 10 dependent survey questions (independent questions were not considered). If the F statistic was greater than or equal to 0.1 the variances of the two groups are assumed equal. The variances are considered). If the F statistic was greater than or equal to 0.1 the

TABLE 5

COMMAND ASSIGNMENT OF RESPONDENTS WHO WORK WITH ADA

Command	*Number of Responses*	*Command*	*Number of Responses*
AFCC	15	AFSPACECOM	5
AFSC	30	HQ/AF	1
AFLC	26	USAFE	1
TAC	9	PACAF	1
MAC	8	ATC	1
SAC	4	AFOTEC	2

variances of the two groups are assumed equal. The variances are assumed unequal otherwise. The t-critical value for an alpha of 0.1 with 192 degrees of freedom (infinity) is 1.645 (Devore, 1987:635). If the t-value is greater than 1.645 or less than -1.645, the Null Hypothesis (the means are equal) is rejected. Table 6 shows the results of the t-test for those questions considered significant. When analyses include the questions in Table 6, a comparison will be made between treatment group and control group responses for those questions.

Analyses to Answer Investigative Questions. The following investigative questions were proposed, via the survey in Appendix B, to Air Force software managers throughout the commands listed in Tables 3 and 4 above. Next is the analysis and the conclusion for these questions.

Investigative Question 1. What formal software education have they received?

Discussion. This question can be answered from responses to survey questions 11 through 23. These questions ask the

TABLE 6

T-TEST RESULTS

Question Number	*Prob > F*	*t*
28	0.0000	-2.0940
40	0.0036	-1.8459
42	0.0003	-4.3099
43	0.0000	-5.7198
44	0.0000	-3.1427
46	0.1042	-2.6020
48	0.8664	-1.9721
49	0.4478	-2.3906
50	0.0511	-2.6964
51	0.5375	-2.7830

respondents about the age of their most current education. The responses indicate that most respondents have had a variety of software education. Table 7 shows the number of respondents who have been educated or trained in the software areas and languages listed.

An important observation here is that approximately 52 percent of the respondents have had no Ada education or training. Of even more importance is that only 60 percent of the people who work with Ada have been educated or trained in Ada.

Conclusion. Overall, a majority of the population has been educated or trained in software engineering in general or software engineering disciplines such as requirements and specifications, maintenance and testing, and design. Most of the population know the benefits of using sound software engineering to develop software but they may not know the benefits of Ada as its companion.

TABLE 7

PERCENTAGE OF RESPONDENTS WHO HAVE HAD SOFTWARE EDUCATION OR TRAINING

Software Area (A) or Language (L)	*Percent of Responses*	*Software Area (A) or Language (L)*	*Percent of Responses*
Programming (A)	97.42	COBOL (L)	52.73
Engineering (A)	78.87	Fortran (L)	83.51
Requirements/Specs (A)	73.20	Pascal (L)	54.64
Maintenance/Testing (A)	69.59	Algol (L)	7.73
Design (A)	72.68	Modula-2 (L)	4.64
Ada (L)	48.45	C (L)	29.90
Jovial (L)	13.92		

Investigative Question 2. Are they knowledgeable about software engineering principles that Ada supports?

Discussion. Survey questions 42 through 48 test the respondents' knowledge about whether or not Ada supports certain general software engineering principles. Ada supports all of the principles the respondents were questioned about (though some software experts argue the degree to which Ada supports the principles). Table 8 shows the percentages of respondents who know that Ada supports the principles.

As Table 7 shows, about 82 percent of the people surveyed have had some formal software engineering education or training while less than 49 percent have been educated in Ada. The low percentages of people who know that Ada supports these principles may either reflect; (1) some of

TABLE 8

PERCENTAGE OF RESPONDENTS WHO KNOW ADA SUPPORTS SOFTWARE ENGINEERING PRINCIPLES

Software Engineering Principle	*Treatment Group*	*Control Group*	*Total Agree*
Abstraction	81.43	58.87	67.01
Information Hiding	88.57	58.07	69.07
Modularity	90.00	76.61	81.44
Localization	60.00	53.23	55.67
Uniformity	68.57	54.03	59.28
Completeness	44.29	41.46	42.49
Confirmability	41.43	30.89	34.72

these principles were not taught to them, (2) due to the time lapse since their education they have forgotten about the topics, (3) they have opinions otherwise, (4) they are not certain, or (5) their education, although hitting on some of the software engineering topics, was not very effective. The statistics show that a considerable percentage of respondents fit the fourth category.

The treatment group (those who certainly have been educated on software engineering at AFIT) showed higher percentages in agreement to the questions. The reason for this disparity may be the maturity of the control group respondents Ada education or training. The treatment group members were surveyed at the end of their software engineering course where these ideas are still fresh.

Conclusion. Overall, only a slight majority of the population are knowledgeable that Ada supports the software engineering principles abstraction, information hiding, modularity, localization, and uniformity. A minority of the population are knowledgeable that Ada supports the software engineering principles completeness and confirmability.

Investigative Question 3. ***Are they aware of the advantages of using Ada as a software engineering tool?***

Discussion. This question is addressed through survey questions 40 and 41. There is exactly one correct answer to each of these questions. Approximately 87 percent of the respondents answered question 40 correctly and approximately 68 percent answered question 41 correctly. The percentages of the treatment group compared to the control group are considerably different for question 40 (92.75 versus 83.87, respectively) and for question 41 (80.00 versus 60.98, respectively). Note here that question 41 did not show significance in the t-test, but, the differences are noteworthy.

Although the percentage of correct responses of the control group respondents is considerably lower than the treatment group, the percentages are somewhat closer and higher (92.11 treatment group, 87.50 control group) among those in the two groups who work with Ada.

Conclusion. Overall, the statistics indicate that a majority of the population knows that the intent of Ada is to enhance software lifecycle benefits and that Ada inherently supports software engineering principles.

Investigative Question 4. ***Do they believe additional knowledge about Ada and software engineering would improve their ability to perform their job?***

Discussion. This question is answered with responses from survey question 49, which addresses the need for a formal course in the Ada programming language, and question 50, which addresses the need for a formal course in software engineering. Table 9 shows the overall responses to question 49.

TABLE 9

RESPONSE PERCENTAGES FOR SURVEY QUESTION 49

Strongly Agree	*Mildly Agree*	*Neutral*	*Mildly Disagree*	*Strongly Disagree*
30.93	27.32	11.86	11.34	18.56

Of the 57 respondents who disagree with question 49, 43 do not work with Ada. The remaining 14 who disagree have either been educated in the language or they work with Ada at a level where they do not need an in-depth knowledge of the language.

The difference in responses of the treatment and control groups is significant for question 49. However, the percentage of those who do not work with Ada are nearly identical. Table 10 compares the responses of the control group, treatment group, and the respondents who work with Ada.

TABLE 10

CONTROL GROUP, TREATMENT GROUP, AND ADA WORKER
RESPONSE PERCENTAGES FOR SURVEY QUESTION 49

Group	*Strongly Agree*	*Mildly Agree*	*Neutral*	*Mildly Disagree*	*Strongly Disagree*
Control	26.61	25.00	13.71	12.90	21.77
Treatment	33.75	31.43	8.57	8.57	12.86
Ada Workers	50.49	31.07	8.74	3.88	5.83

Table 11 shows the overall responses to question 50 which addresses the need for a formal course in software engineering.

TABLE 11

RESPONSE PERCENTAGES FOR SURVEY QUESTION 50

Strongly Agree	*Mildly Agree*	*Neutral*	*Mildly Disagree*	*Strongly Disagree*
57.73	25.77	6.70	4.64	5.15

Of the 17 respondents who disagree with question 50, all but 3 have had some form of formal software engineering education. There is no explanation why the 3 respondents disagree.

The difference in responses of the treatment and control groups is also significant for question 50. Approximately 89 percent of the treatment group agreed while approximately 81 percent of the control group agreed. This difference may be explained by the fact that all treatment group respondents had completed the AFIT SPDP *Software Engineering Concepts Course* and therefore realize the positive impact the education can have on job performance.

Conclusion. A majority of software managers feel that formal courses in the Ada programming language and/or software engineering would enhance their ability to perform their job.

Investigative Question 5. What formal education would enhance their ability to perform their job?

Discussion. This question can be answered from responses to survey questions 49, 50, and 51. These three survey questions address the need for formal education in Ada, software engineering, and domain specific software applications. Table 12 shows the percentage of responses to these questions.

TABLE 12

FORMAL EDUCATION NEEDS

Topic	*Strongly Agree*	*Mildly Agree*	*Neutral*	*Mildly Disagree*	*Strongly Disagree*
Ada	30.93	27.32	11.86	11.34	18.56
Software Engineering	57.73	25.77	6.70	4.64	5.15
Domain Application	40.41	33.16	18.65	4.66	3.11

The significance between the treatment group and control group responses for questions 49 and 50 was discussed for investigative question 5. The t-test also shows that question 51 has significant response differences. Table 13 shows the responses of the two groups.

TABLE 13

CONTROL GROUP AND TREATMENT GROUP
RESPONSE PERCENTAGES FOR SURVEY QUESTION 51

Group	*Strongly Agree*	*Mildly Agree*	*Neutral*	*Mildly Disagree*	*Strongly Disagree*
Control	31.71	37.39	21.14	6.50	3.25
Treatment	55.71	25.71	14.29	1.43	2.86

This disparity seems to be a function of the number of neutral responses rather than the number of negative responses. The treatment group, more currently educated in software engineering principles, are made aware of the potential differences among domain applications. Some of the control group members may not be aware of these differences.

Conclusion. As the above data reflects, a majority of the population feels that they could benefit from formal courses in each topic. Ada may be less desired because, as mentioned earlier, only about 47 percent of the respondents work with Ada. Because of the population definition, it should be expected that most respondents would require software engineering education. This research confirms this expectation with 83.50 percent agreeing that a formal course would help them. Also, because of the diversity of software jobs in the Air Force, it should be expected that most of the population would desire a domain applications course. Again, the 73.57 percentage of respondents in agreement confirm this expectation.

Investigative Question 6. How often should the formal education occur?

Discussion. This question requires analysis of survey questions 24, 25, 26, 27, 28, 29, 30, 32, and 33. Questions 24 through 26 address general education, questions 27 through 30 address domain specific education, and questions 32 and 33 provide data to support questions 24 through 30.

In reference to general education for Ada or software engineering, 19.59 percent of the respondents feel that they should be given only once, 76.80 percent feel that they should be given periodically, and 4.12 feel they are not necessary at all. (These percentages do not add to 100 because 4 respondents answered that general education should be given both once and periodically.)

In reference to domain applications education for Ada and software engineering, 6.70 percent of the respondents feel they should be given once only, 82.99 percent feel they should be given on an as needed basis, 65.98 believe they should be given periodically, and 3.63 percent feel they should not be given.

Question 32 addresses the need for Ada and software engineering on a domain level. The response (77.72 percent of the respondents) to this question supports an inference that software managers feel a need for both general and domain specific education.

Question 33 addresses the need for Ada and software engineering education based on a person's grade or position. The response to this question was distributed almost evenly. Here, 39.90 percent believe education should be tailored to rank or position, 26.94 percent believe managers should receive the same education regardless of rank, and 27.46 percent believe managers should receive education both general in nature and tailored to their rank or position.

Question 33 correlates with question 28 which says that domain specific education should be given as needed. As managers change rank or position, they usually change work domains. It follows that because promotions or job changes do not occur annually or periodically, the education would be given as needed.

The t-test results indicate that the control group responded significantly different than treatment group to the need for as-need domain specific education. Table 14 shows the responses of the two groups.

TABLE 14

CONTROL GROUP AND TREATMENT GROUP
RESPONSE PERCENTAGES FOR SURVEY QUESTION 28

Group	*Agree*	*Disagree*	*Undecided*
Control	79.84	12.90	7.26
Treatment	88.57	10.00	1.43

As with survey question 51, the difference noted here is likely to be the result of the fact that the treatment group is made aware of domain differences through the SPDP course. The control group, on the other hand, may not be aware of the differences.

Conclusion. A majority of the population believe that general education necessary to support the DOD policy should be given periodically and that domain specific education should be given periodically or as-needed.

Investigative Question 7. Who should fund the education?

Discussion. Survey question 31 addresses this question. The respondents were given the flexibility to choose one funding agency or several. There was no overwhelming majority responses for this question, however, the most frequent responses indicate that the owning command (20.83 percent) or Air Training Command (ATC) (23.44 percent) should provide funding for education in support of the DOD policy.

Rather than indicate one agency, many respondents chose to offer other possibilities and mixtures of funding sources. The most common mix here was that the owning command should pay for domain specific education and ATC should pay for general education. Some go even further and state that project related education should be funded by project funds.

Conclusion. There is no clear indication here as to what single agency should provide the funds to educate software managers.

Regression Analysis Results. A stepwise regression was performed to test the linear relationship between the criteria (dependent variable) and a set of predictor (independent) variables to determine the 'best' education model for software managers. The two types of education covered in the survey are (1) Ada and (2) software engineering. The questions to answer for each type are what type of education (domain specific and/or general) is needed, at what depth (by rank and/or position) the education should be taught, and when (not at all, periodically, as needed, or once only) the education should be given.

The descriptive statistics show that both general and domain specific education are necessary. The statistics show that the education should be taught with both regard to rank and/or position and without regard to rank and/or position (depth). The statistics also show that general education should be given periodically and domain education should be given as needed or periodically. An insignificant amount of the population believe that education should be given only once or not at all.

Ada Education Model Analysis. As mentioned, the criteria for the Ada education model are survey questions 25 (PGEPER), 28 (PDEAN), 29 (PDEPER), 32 (PEDOMAIN), 33 (PEGRADE), 50 (SFA), and 51 (SDAPP). The predictors are the survey questions grouped under SE_AREAS and ADA_JOBS (see Table 1), and questions 1 (GRADE), 2 (CMND), 3 (AFSC), 4 (JYEARS), 5 (JPM), 6 (JTM), 7 (JDEV), 8 (JACQ), 9 (JMX), 10 (JOTHER), 11 (ECP), 12 (ESE), 16 (EADA), 17 (EJOV), 18 (ECOB), 19 (EFORT), 20 (EPASC), 21 (EALG), 22 (EMOD2), 23 (EC), AND 39 (AJOTHER) (survey questions not grouped). Reference Appendices B or C for a description of the questions.

Criterion and Predictor Analysis. SAS performs the regression for a significance level (the probability that F is greater than F-critical) of 0.15. Any predictor not fitting this criteria is not included as a predictor to the model. Table 15 shows the model unique criteria and significant predictors.

TABLE 15

ADA UNIQUE CRITERIA AND SIGNIFICANT PREDICTORS

Criteria	*Predictor(s)*	*Prob>F*
SFA	ADA_JOBS	0.0001
	CMND	0.0365
	ECP	0.1058
	JMX	0.0555

Table 16 shows the common model criterion and the predictors that fit the required significance level.

The stepwise regression analysis shows that the CMND (command related) is a common predictor to SFA (a need for Ada education) and PGEPER (the need for periodic general education). This indicates that the need for periodic general education in Ada depends on the command to which the personnel are assigned. This makes sense because not all commands deal with Ada at the working level. For instance, Systems

TABLE 16

COMMON CRITERION AND SIGNIFICANT PREDICTORS

Criteria	*Predictor(s)*	*Prob>F*
PGEPER	CMND	0.0240
	EMOD2	0.1474
PDEPER	EC	0.0292
	JDEV	0.0393
	AFSC	0.0455
	JPM	0.0730
	JACQ	0.0523
PDEAN	EALG	0.0673
PEDOMAIN	JOTHER	0.0535
	JPM	0.0549
	ADA_JOBS	0.1360
PEGRADE	GRADE	0.0029
	JMX	0.0208
	EFORT	0.0449
	ESE	0.0777
	JOTHER	0.0933
SDAPP	ECOB	0.0051
	JTM	0.0175
	EMOD2	0.0452
	JDEV	0.0484
	JMX	0.0272
	EC	0.1049
SFSE	ADA_JOBS	0.0258
	JMX	0.0005
	ECOB	0.1181
	EMOD2	0.1486
	EFORT	0.1378
	ESE	0.1406

Command deals with software acquisition and therefore would not need more than a general knowledge of Ada. In contrast, using commands such as MAC, TAC, and SAC actually deal with Ada at the code level. They would require a much more in-depth knowledge of Ada.

The predictor ADA_JOBS (a group of different jobs that deal with Ada) is common to SFA, PEDOMAIN (need for domain education), and SFSE (need for software engineering education). This indicates that the need for general and domain specific education in Ada and software engineering depends on whether or not the person works with Ada. As discussed in Chapter 2, using Ada under the auspices of sound software engineering principles provides a cost effective, high quality product.

The predictor JMX (software maintenance/testing job) is common to SFA, SFSE, SDAPP (a need for domain applications education), and PEGRADE (a need for education based on rank or position). This indicates that the need for different levels of Ada and software engineering education with regard to domain and rank and/or position depends mostly on the software maintenance/testing function. Again, because maintenance personnel work with Ada at the code level, they may need different levels of education within their rank structure (organizational hierarchy). Of course, people who perform the maintenance/testing function would need the same level of software engineering education, independent of the language.

ECP (people educated in computer programming) is the only predictor unique to SFA. As the descriptive statistics show, a majority of the population have been educated in computer programming (language unspecified). Apparently, a significant portion of these personnel program in Ada and would benefit from a formal course on Ada programming.

V. Conclusions and Recommendations

This chapter formulates conclusions about the analyses performed on the investigative questions in Chapter 4 to answer the management question proposed in Chapter 1. Recommendations are also included for further research in the area of this study.

Management Question

The management question was concerned with whether or not software managers need more education about the Ada programming language and software engineering. If so, what kind of education is needed and how often should it be given?

Do software managers need more education about Ada and software engineering? The statistics and analyses show that many software managers believe they need one or the other, or both. Of the people working in software functions in the Air Force, less than 50 percent have been educated in Ada and from 70 to 79 percent have had some form of software engineering education.

The education should not be generic across-the-board. Rather, the type, frequency, depth, or breadth of education depends on combinations of the individual's grade, position, previous education, and work domain (for instance, command and job type).

This study indicates that the people assigned to commands such as MAC, SAC, and TAC would benefit most from a programming-level knowledge of Ada and software engineering. Their duties often include the code-level functions of incorporating software changes and corrections or design recovery and software redesign for existing software. Without a working knowledge of Ada and software engineering, their tasks would be time consuming and inefficient.

In contrast, people who work in software specification and acquisition functions usually work with Ada, or software in general, at a high level and would not benefit from a working level knowledge of Ada or software engineering. Here, an overview of Ada plus an executive-level course may be appropriate. This level of education would provide enough detail to enable these people to knowledgeably acquire Ada systems.

People who do not currently work with Ada would not benefit from a formal Ada programming course until they are ready to use it. However, these people may benefit from a formal software engineering course. Again, the depth of the education is a function of the person's duties.

As with education type and depth, software managers also need different frequencies of Ada and software engineering education. The study indicates that the frequency is primarily a function of job type (duties and specialty/job codes) and command. The two frequencies considered are: 1) periodic and 2) as-needed. Although they may seem the same they are not. Here, periodic education would likely consist of annual courses to update software managers on new technologies and methods. As-needed education may be given to upgrade a person's knowledge because of a permanent change of station (PCS) or a change in their grade or position (at which time their job type may change). Whether or not a person works with Ada has little or no impact on the needed frequency of education.

This study also shows a need for domain specific education in both Ada and software engineering. Because people do not change domains frequently however, this education would occur on an as-needed basis. As with Ada and software engineering, domain education could be designed with both depth and breadth in mind.

Conclusions

The conclusion is that both Ada and software engineering education are needed if software personnel are expected to work effectively within the bounds of the DOD Ada policy. The education however, should not be a one-time, general overview for all. Instead, the depth, breadth, and frequency should be tailored to meet the needs of different software functions.

Overall, the population of software managers seem to be fairly well versed in software engineering principles and Ada's support of these principles. However, their knowledge level may not be sufficient to identify the inherent properties of these principles in a software product. In short, they may only be 'Buzz Words'.

Recommendations For Further Research

The data collected for this thesis would support a greater depth of research about the educational needs of Air Force software managers. The data exists to study the needs of individual grades (company grade officers versus field grade officers), commands (user commands versus others), specialty codes (software personnel versus acquisition personnel), or job types (maintenance needs versus design needs). Further studies may result in full education models for Air Staff and individual organizations to use in educating Air Force software personnel.

This research brought out the desires of software managers for domain specific education. Further research could include identifying how many domains exist and to what level of specificity domains need to be taught. The research could also determine who is qualified to provide the education.

To assure software personnel are getting the proper education, Air Staff might investigate implementing a certification program for software personnel like the program established for acquisition managers. This would facilitate an accumulation of knowledge commensurate with the individual's grade or position.

In summary, if people are expected to support policies, they should be prepared to do it. Education is one means to assure the people have the right tools to make prudent decisions. The short-run cost of educating a person who makes multi-million dollar decisions will surely be recouped in the long-run with profits from lifecycle savings.

Appendix A: Glossary of Terms

Acronyms

AFCC. Air Force Communications Command

AFIT. Air Force Institute of Technology

AFLC. Air Force Logistics Command

AFMPC. Air Force Manpower and Personnel Center

AFSC. Air Force Systems Command

AFSPACECOM. Air Force Space Command

AJPO. Ada Joint Program Office

ASEET. Ada Software Engineering Education and Training

ATC. Air Training Command

BAR. Broad Area Review

DOD. Department of Defense

HQAF. Headquarters Air Force

MAC. Military Airlift Command

PACAF. Pacific Air Command

SAC. Strategic Air Command

SPDP. Software Professional Development Program

TAC. Tactical Air Command

USAFE. United States Air Forces in Europe

Software Related Terms

Abstraction.

(1) A view of a problem that extracts the essential information relevant to a particular purpose and ignores the remainder of the information. (International, 1983)

(2) The process of forming an abstraction. (International, 1983)

Ada. The standard programming language for Mission Critical computer systems in the DoD. Currently the language is defined in ANSI/MIL-STD-1815A.

Adaptive Maintenance. Maintenance performed to make a software product usable in a changed environment (International, 1983)

Attribute. For purposes of clarification and prioritization, attributes are characteristics of the functions desired by the client.

Class. A set of objects that share a common structure and a common behavior. (Booch, 1991:513)

Cohesion. A measure of how closely related the tasks performed by a specific module are. (Davis, 1990:373)

Completeness. The property of software where the interface of the software class or module captures all of the meaningful characteristics of the abstraction. (Booch, 1991:125)

Confirmability. Decomposition facilitating testing.

Corrective Maintenance. Maintenance performed specifically to overcome existing faults. (International, 1983)

Design. The period of time in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements. (International, 1983)

Domain Analysis. Domain analysis defines the common characteristics of the environment a family of similar systems will operate in.

Embedded system. A computer system that is part of a larger system that is not specifically designed for computation.

Hardware. Physical equipment used in data processing, as opposed to computer programs, procedures, rules, and associated documentation. Contrast with Software. (International, 1983)

Information Hiding. The process of hiding all of an object's details that do not contribute to its essential characteristics. (Booch, 1991:514)

Localization. Physical proximity.

Management. All the activities and tasks undertaken by one or more persons for the purpose of planning and controlling the activities of others in order to achieve an objective or complete an activity that could not be achieved by the others acting alone.

Method. A systematic procedure, technique, or mode of enquiry employed by or proper to a particular discipline or art. (American, 1979)

Modularity. The property of a system that has been decomposed into a set of cohesive and loosely coupled modules. (Booch, 1991:515)

Perfective Maintenance. Maintenance performed to improve performance, maintainability, or other software attributes. (International, 1983)

Project Management. A system of procedures, practices, technologies, and know-how that provides the planning, organizing, staffing, directing, and controlling necessary to successfully manage an engineering project.

Software Engineering. The systematic approach to the development, operation, maintenance, and retirement of software. (International, 1983)

The application of common sense, mathematical principles, and engineering discipline to developing solutions for software problems that don't violate common sense.

Software Lifecycle. The span of time from the concept of a software product to the time the software is destroyed of permanently retired.

Software Maintenance. Modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment. (International, 1983)

Software Process.

(1) The set of tools, methods, and practices used to produce a software product.

(2) The technical and management framework established for applying tools, methods, and people to the software task.

Standards. A set of rules that attempt to establish or enforce a specific style.

Structured programming. A programming methodology for generating structured code. A product is structured if the code blocks are connected only by concatenation, selection, iteration, or a call, and every block has exactly one entry and one exit.

System.

(1) A collection of people, machines, and methods organized to accomplish a set of specific functions. (International, 1983)

(2) An integrated whole that is composed of diverse, interacting, specialized structures and subfunctions. (International, 1983)

(3) A group or subsystem united by some interaction or interdependence, performing many duties but functioning as a single unit. (International, 1983)

(4) Production of a single set of optimum outputs from the given set of inputs, with respect to some appropriate measure of effectiveness

System Development Process. The process of delivering to the users what they want and delivering to the maintainers a system they can maintain.

System Testing. The process of testing an integrated hardware and software system to verify that the system meets its specified requirements. (International, 1983)

Uniformity. The property where software modules use consistent notation.

Statistical Terms

Reliability. An estimate of the degree to which a measurement is free of random or unstable error. (Emory, 1985:98)

Statistical Confidence. The probability that in an experiment, the true mean will be contained in the defined interval. (Devore, 1987:254)

Stratified. The division of a population to be sampled into blocks, each of which is sampled separately. (Meyburg and Stopher, 1979:30)

Validity. Indicates the degree to which an instrument measures what it is supposed to measure. (Emory, 1985:109)

Appendix B: Questionnaire and Instructions

USAF Survey Control No. 91-25

QUESTIONNAIRE INSTRUCTIONS

PLEASE READ THE FOLLOWING INSTRUCTIONS COMPLETELY BEFORE BEGINNING THE QUESTIONNAIRE.

1. Answer ALL questions.

2. Use a pencil to record your answers (No. 2 or equivalent). NO INK.

QUESTIONS ARE NUMBERED, RESPONSES ARE ALPHABETIZED

3. Record your responses on the accompanying answer form by completely darkening the circles that correspond to your response.

4. DO NOT FOLD THE ANSWER FORM AND DO NOT SEPARATE THE PAGES OF THE ANSWER FORM.

5. Ignore the Identification box on the first page, upper left corner of the answer form.

6. Complete "other" responses in the blank area on the last (4th) page of the answer form. Be sure to number the response with the corresponding question number.

EXAMPLE: 2. USAFE
10. Programming

7. Possible multiple answer questions.

QUESTION 10. If the response is not *a. 0%*, complete the "other" response as instructed in 6 above.

QUESTION 39. If the response is *b. Yes*, complete the "other" response as instructed in 6 above.

8. If you wish to provide comments other than described in 6 above, write the answer form number (located on the first page, bottom right corner of the form) on any additional comment sheet(s).

9. Return ONLY the answer form and additional comments in the pre-addressed envelope.

10. THANK YOU FOR YOUR PARTICIPATION.

I. *This section applies to your current job*.

1. Grade/Rank a. 0-3 c. 0-5 e. GS-13
b. 0-4 d. GS-12 f. GS-14 g. Other ____

2. To what Air Force command are you currently assigned?

a. AFCC c. AFLC e. MAC g. AFSPACECOM
b. AFSC d. TAC f. SAC h. Other ________

3. What is your primary Air Force specialty code? (Military write 3rd character in comments.)

a. 26___X c. 28___X e. 334 g. 856
b. 27___X d. 49___X f. 855 h. Other ________

4. How many years have you been in your current job?

a. Under 2 b. 2-5 c. 6-10 d. More than 10

What percentage of your current job involves the following software functions:

a. 0% (or None) b. 1-25% c. 26-50% d. 51-75% e. 76-100%

5. Program management.
6. Technical management.
7. Development.
8. Acquisition.
9. Maintenance/testing
10. Other ___________

II. *This section applies to your education/training background*.

Identify which of the following topics you have received formal education/training in (most current) by indicating how many years ago you received it.

YEARS: a. No education/training b. Less than 5 c. 5-10
d. more than 10

11. Computer programming.
12. Software engineering.
13. Software requirements/specification.
14. Software maintenance/testing.
15. Software design.

Identify which of the following programming languages you have had formal education/training in by indicating how many years ago you received it.

YEARS: *a. No education/training* *b. Less than 5* *c. 5-10* *d. more than 10*

16. Ada	18. COBOL	20. Pascal	22. Modula-2
17. Jovial	19. Fortran	21. Algol	23. C

III. <u>This section applies to the current Department of Defense policy that mandates using the Ada programming language for all DOD software system developments</u>.

General education to support this policy should be given ...

a. Agree *b. Disagree* *c. Undecided*

24. Once only
25. Periodically (eg., annually, bi-annually)
26. Not needed

Domain specific education to support this policy should be given ...

a. Agree *b. Disagree* *c. Undecided*

27. Once only
28. As needed (eg., when unit objectives change)
29. Periodically
30. Not needed

31. From where should funds for the education come? If you believe more than one response is appropriate, choose f and explain in the comments.

a. Unit training funds
b. Project funds
c. Owning Command
d. Air Training Command
e. Air University
f. Other ______
g. Undecided

32. Software managers should ...

a. receive software education specific to their application domain.
b. receive general software education regardless of their application domain.
c. both a & b.
d. Undecided.

33. Software managers should ...

a. receive a depth of education tailored to their rank/grade and/or position.
b. receive the same depth of education regardless of their rank/grade and/or position.
c. both a & b.
d. Undecided.

IV. *This section addresses the Ada programming language and software engineering principles. Please respond to the best of your ability.*

Does your current job involve the *Ada programming language* in any of the following capacities? (Regardless of the time spent on the task.)

a. Yes *b. No*

34. Program management	36. Development	38. Maintenance/testing
35. Technical management	37. Acquisition	39. Other ____________

40. Which statement below BEST explains the Government's reason for developing the Ada programming language.

a. Language standardization to enhance software lifecycle benefits.
b. Because Ada's performance is superior to all other high-level languages.
c. Because Ada is easier to program than other languages.
d. Uncertain.

41. A major benefit of the Ada programming language is ....

a. its ability to support all known types of programming applications.
b. its simplicity to program.
c. its inherent support for software engineering principles.
d. Uncertain.

Ada supports the following software engineering principles?

a. Yes *b. No* *c. Uncertain*

42. Abstraction	44. Modularity	46. Uniformity
43. Information hiding	45. Localization	47. Completeness
		48. Confirmability

Answer the remaining questions according to the following scale.

a. Strongly Agree b. Mildly Agree c. Neutral d. Mildly Disagree e. Strongly Disagree

49. A formal course on Ada programming would enhance my ability to perform my current job.

50. A formal course on software engineering would enhance my ability to perform my current job.

51. A domain software applications course would enhance my ability to perform my current job.

Appendix C: Description of Survey Question Names

This appendix contains a brief description of each survey question and includes the names or abbreviations (numerical and alphabetical) assigned to the questions. The format is as follows: question number (question name): description.

Numerical Listing

Question 1 (GRADE): The grade or rank of the respondent.

Question 2 (CMND): The command to which the respondent is assigned.

Question 3 (AFSC): The Air Force Specialty Code or job description number of the respondent.

Question 4 (JYEARS): The number of years the respondents have been in their current job.

Question 5 (JPM): How much of the respondent's job includes software program management.

Question 6 (JTM): How much of the respondent's job includes software technical management.

Question 7 (JDEV): How much of the respondent's job includes software development.

Question 8 (JACQ): How much of the respondent's job includes software acquisition.

Question 9 (JMX): How much of the respondent's job includes software maintenance/testing.

Question 10 (JOTHER): How much of the respondent's job includes software functions not defined in questions 5 through 9.

Question 11 (ECP): The age of the respondent's most current education or training in computer programming.

Question 12 (ESE): The age of the respondent's most current education or training in software engineering.

Question 13 (ESREQ): The age of the respondent's most current education or training in software requirements/specification.

Question 14 (ESMX): The age of the respondent's most current education or training in software maintenance/testing.

Question 15 (ESDES): The age of the respondent's most current education or training in software design.

Question 16 (EADA): The age of the respondent's most current education or training in the Ada programming language.

Question 17 (EJOV): The age of the respondent's most current education or training in the Jovial programming language.

Question 18 (ECOB): The age of the respondent's most current education or training in the COBOL programming language.

Question 19 (EFORT): The age of the respondent's most current education or training in the Fortran programming language.

Question 20 (EPASC): The age of the respondent's most current education or training in the Pascal programming language.

Question 21 (EALG): The age of the respondent's most current education or training in the Algol programming language.

Question 22 (EMOD2): The age of the respondent's most current education or training in the Modula-2 programming language.

Question 23 (EC): The age of the respondent's most current education or training in the C programming language.

Question 24 (PGEONE): General education to support the DOD policy should be given once only.

Question 25 (PGEPER): General education to support the DOD policy should be given periodically.

Question 26 (PGENO): General education to support the DOD policy is not needed.

Question 27 (PDEONE): Domain specific education to support the DOD policy should be given once only.

Question 28 (PDEAN): Domain specific education to support the DOD policy should be given as needed.

Question 29 (PDEPER): Domain specific education to support the DOD policy is not needed.

Question 30 (PDENO): Domain specific education to support the DOD policy is not needed.

Question 31 (PFUNDS): Identifies funding sources for the education.

Question 32 (PEDOMAIN): Identifies what education software managers should receive in reference to their domain.

Question 33 (PEGRADE): Identifies the depth of education software managers should receive in reference to their grade/rank and/or position.

Question 34 (AJPM): How much of the respondent's job includes Ada software program management.

Question 35 (AJTM): How much of the respondent's job includes Ada software technical management.

Question 36 (AJDEV): How much of the respondent's job includes Ada software development.

Question 37 (AJACQ): How much of the respondent's job includes Ada software acquisition.

Question 38 (AJMX): How much of the respondent's job includes Ada software maintenance/testing.

Question 39 (AJOTHER): How much of the respondent's job includes Ada software functions not defined in questions 34 through 38.

Question 40 (AREASON): Identifies why the Government developed the Ada programming language.

Question 41 (ABENEFIT): Identifies a major benefit of the Ada programming language.

Question 42 (AABSTR): The software engineering principle 'abstraction'.

Question 43 (AIH): The software engineering principle 'information hiding'.

Question 44 (AMOD): The software engineering principle 'modularity'.

Question 45 (ALOCAL): The software engineering principle 'localization'.

Question 46 (AUNIF): The software engineering principle 'uniformity'.

Question 47 (ACOMP): The software engineering principle 'completeness'.

Question 48 (ACONF): The software engineering principle 'confirmability'.

Question 49 (SFA): Identifies the desire for a formal Ada programming course.

Question 50 (SFSE): Identifies the desire for a formal software engineering course.

Question 51 (SDAPP): Identifies the desire for a formal domain software applications course.

Alphabetical Listing

AABSTR	(42)
ABENEFIT	(41)
ACOMP	(47)
ACONF	(48)
AFSC	(3)
AIH	(43)
ALOCAL	(45)
AMOD	(44)
AJACQ	(37)
AJDEV	(36)
AJMX	(38)
AJOTHER	(39)
AJPM	(34)
AJTM	(35)
AREASON	(40)
AUNIF	(46)
CMND	(2)
EADA	(16)
EALG	(21)
EC	(23)
ECOB	(18)
ECP	(11)
EFORT	(19)
EJOV	(17)
EMOD2	(22)
EPASC	(20)
ESE	(12)
ESDES	(15)
ESMX	(14)
ESREQ	(13)
GRADE	(1)
JACQ	(8)
JDEV	(7)
JMX	(9)
JOTHER	(10)
JPM	(5)
JTM	(6)
JYEARS	(4)
PDEAN	(28)
PDENO	(30)
PDEONE	(27)
PDEPER	(29)
PEDOMAIN	(32)
PEGRADE	(33)
PFUNDS	(31)
PGEONE	(24)
PGENO	(26)
PGEPER	(25)
SDAPP	(51)
SFA	(49)
SFSE	(48)

Appendix D: SAS Programs

SAS Program to Determine Dependent Variables

```
options linesize=80;
data temp;

    infile research missover;

    input  form $5-8 pgeone 32 pgeper 33 pgeno 34 pdeone 35 pdean 36
            pdeper 37 pdeno 38 pfunds 39 pedomain 40 pegrade 41
            areason 48 abenefit 49 aabstr 50 aih 51 amod 52 alocal 53
            aunif 54 acomp 55 aconf 56 sfa 57 sfse 58 sdapp 59;

proc factor r=varimax;

proc corr alpha;
     var pgeone pdeone pgeno pdeno;
proc corr alpha;
     var pgeper pdeper;
proc corr alpha;
     var pedom pegrade pdean;
proc corr alpha;
     var areason abenefit;
proc corr alpha;
     var aabstr aih amod alocal aunif acomp aconf;
proc corr alpha;
     var sfa sfse sdapp;
```

SAS Program to Determine Independent Variables

```
options linesize=80;
data temp;

    infile research missover;

    input  form 1-8 grade 9 cmnd 10 afsc 11 jyears 12 jpm 13 jtm 14
            jdev 15 jacq 16 jmx 17 jother 18 ecp 19 ese 20 esreq 21
            esmx 22 esdes 23 eada 24 ejov 25 ecob 26 efort 27 epasc 28
            ealg 29 emod2 30 ec 31 ajpm 42 ajtm 43 ajdev 44 ajacq 45
            ajmx 46 ajother 47;

proc factor r=varimax;

proc corr alpha;
      var ajpm ajtm ajdev ajacq ajmx;
proc corr alpha;
      var esreq esmx esdes;
```

SAS Program to Determine Descriptive Statistics and Response Tables

```
options linesize=80;
data temp;

    infile research missover;

    input  form 5-8 grade 9 cmnd 10 afsc 11 jyears 12 jpm 13 jtm 14
            jdev 15 jacq 16 jmx 17 jother 18 ecp 19 ese 20 esreq 21
            esmx 22 esdes 23 eada 24 ejov 25 ecob 26 efort 27 epasc 28
            ealg 29 emod2 30 ec 31 pgeone 32 pgeper 33 pgeno 34
            pdeone 35 pdean 36 pdep 37 pdeno 38 pfunds 39 pedomain 40
            pegrade 41 ajpm 42 ajtm 43 ajdev 44 ajacq 45 ajmx 46
            ajother 47 areason 48 abenefit 49 aabstr 50 aih 51 amod 52
            alocal 53 aunif 54 acomp 55 aconf 56 sfa 57 sfse 58
            sdapp 59;

proc format print;
      value formfmt  0001-0070='treatment'
                     0101-0225='control';

proc means;
      var grade--sdapp;
proc freq;
      format form formfmt.;
      tables form*(grade--sdapp);
```

SAS Program For t-test Analysis

```
options linesize=80;
data temp;

    infile research missover;

    input  form 5-8 grade 9 cmnd 10 afsc 11 jyears 12 jpm 13 jtm 14
            jdev 15 jacq 16 jmx 17 jother 18 ecp 19 ese 20 esreq 21
            esmx 22 esdes 23 eada 24 ejov 25 ecob 26 efort 27 epasc 28
            ealg 29 emod2 30 ec 31 pgeone 32 pgeper 33 pgeno 34
            pdeone 35 pdean 36 pdep 37 pdeno 38 pfunds 39 pedomain 40
            pegrade 41 ajpm 42 ajtm 43 ajdev 44 ajacq 45 ajmx 46
            ajother 47 areason 48 abenefit 49 aabstr 50 aih 51 amod 52
            alocal 53 aunif 54 acomp 55 aconf 56 sfa 57 sfse 58
            sdapp 59;

proc format print;
    value formfmt 0001-0080='treatment'
                  0101-0250='control';

proc ttest;
      class form;
      format form formfmt.;
      var form--sdapp;
```

SAS Program For Stepwise Regression Analysis

```
options linesize=80;
data temp;

    infile research missover;

    input  form 5-8 grade 9 cmnd 10 afsc 11 jyears 12 jpm 13 jtm 14
           jdev 15 jacq 16 jmx 17 jother 18 ecp 19 ese 20 esreq 21
           esmx 22 esdes 23 eada 24 ejov 25 ecob 26 efort 27 epasc 28
           ealg 29 emod2 30 ec 31 pgeone 32 pgeper 33 pgeno 34
           pdeone 35 pdean 36 pdep 37 pdeno 38 pfunds 39 pedomain 40
           pegrade 41 ajpm 42 ajtm 43 ajdev 44 ajacq 45 ajmx 46
           ajother 47 areason 48 abenefit 49 aabstr 50 aih 51 amod 52
           alocal 53 aunif 54 acomp 55 aconf 56 sfa 57 sfse 58
           sdapp 59;


/*  INDEPENDENT VARIABLE ASSIGNMENT  */

     ada_jobs=ajpm+ajtm+ajdev+ajacq+ajmx;
     se_areas=esreq+esmx+esdes;

/* ADA Model */

proc stepwise;
     model pgeper=grade cmnd afsc jyears jpm jtm jdev jacq jmx jother
           ecp ese eada ejov ecob efort epasc ealg emod2 ec ajother
           ada_jobs se_areas;

proc stepwise;
     model pdeper=grade cmnd afsc jyears jpm jtm jdev jacq jmx jother
           ecp ese eada ejov ecob efort epasc ealg emod2 ec ajother
           ada_jobs se_areas;

proc stepwise;
     model pdean=grade cmnd afsc jyears jpm jtm jdev jacq jmx jother
           ecp ese eada ejov ecob efort epasc ealg emod2 ec ajother
           ada_jobs se_areas;

proc stepwise;
     model pedomain=grade cmnd afsc jyears jpm jtm jdev jacq jmx jother
           ecp ese eada ejov ecob efort epasc ealg emod2 ec ajother
           ada_jobs se_areas;

proc stepwise;
     model pegrade=grade cmnd afsc jyears jpm jtm jdev jacq jmx jother
           ecp ese eada ejov ecob efort epasc ealg emod2 ec ajother
           ada_jobs se_areas;
```

```
proc stepwise;
     model sfa=grade cmnd afsc jyears jpm jtm jdev jacq jmx jother
           ecp ese eada ejov ecob efort epasc ealg emod2 ec ajother
           ada_jobs se_areas;

proc stepwise;
     model sfse=grade cmnd afsc jyears jpm jtm jdev jacq jmx jother
           ecp ese eada ejov ecob efort epasc ealg emod2 ec ajother
           ada_jobs se_areas;

proc stepwise;
     model sdapp=grade cmnd afsc jyears jpm jtm jdev jacq jmx jother
           ecp ese eada ejov ecob efort epasc ealg emod2 ec ajother
           ada_jobs se_areas;
```

Appendix E: Responses to Survey Questions and Descriptive Statistics

TABLE 17

RESPONSES TO SURVEY QUESTIONS

QUESTION 1 (GRADE)

Group\Choice	a	b	c	d	e	f	g
Treatment	25	2	1	32	5	1	4
Control	49	3	4	25	24	4	15
Total	74	5	5	57	29	5	19

QUESTION 2 (CMND)

Group\Choice	a	b	c	d	e	f	g	h
Treatment	12	19	21	4	2	4	1	7
Control	17	23	43	8	16	11	4	2
Total	29	42	64	12	18	15	5	9

QUESTION 3 (AFSC)

Group\Choice	a	b	c	d	e	f	g	h
Treatment	1	1	10	17	19	18	1	3
Control	1	4	11	41	34	18	0	11
Total	2	5	21	58	53	36	1	14

QUESTION 4 (JYEARS)

Group\Choice	a	b	c	d
Treatment	20	37	9	4
Control	37	61	14	12
Total	57	98	23	16

QUESTION 5 (JPM)

Group\Choice	a	b	c	d	e
Treatment	23	31	8	7	1
Control	38	55	13	16	2
Total	61	86	21	23	3

QUESTION 6 (JTM)

Group\Choice	a	b	c	d	e
Treatment	15	29	16	8	2
Control	26	52	28	10	8
Total	41	81	44	18	10

QUESTION 7 (JDEV)

Group\Choice	a	b	c	d	e
Treatment	26	25	9	6	4
Control	31	58	16	10	8
Total	57	83	25	16	12

QUESTION 8 (JACQ)

Group\Choice	a	b	c	d	e
Treatment	29	20	11	4	6
Control	52	47	17	3	4
Total	81	67	28	7	10

QUESTION 9 (JMX)

Group\Choice	a	b	c	d	e
Treatment	20	29	9	8	4
Control	30	54	17	16	7
Total	50	83	26	24	11

QUESTION 10 (JOTHER)

Group\Choice	a	b	c	d	e
Treatment	46	12	2	0	2
Control	65	17	11	8	4
Total	111	29	13	8	6

QUESTION 11 (ECP)

Group\Choice	a	b	c	d
Treatment	2	43	18	7
Control	3	68	34	19
Total	5	111	52	26

QUESTION 12 (ESE)

Group\Choice	a	b	c	d
Treatment	4	63	3	0
Control	37	77	8	2
Total	41	140	11	2

QUESTION 13 (ESREQ)

Group\Choice	a	b	c	d
Treatment	17	44	8	1
Control	35	67	19	3
Total	52	111	27	4

QUESTION 14 (ESMX)

Group\Choice	a	b	c	d
Treatment	17	48	2	3
Control	42	58	17	7
Total	59	106	19	10

QUESTION 15 (ESDES)

Group\Choice	a	b	c	d
Treatment	20	37	11	2
Control	33	61	25	5
Total	53	98	36	7

QUESTION 16 (EADA)

Group\Choice	a	b	c	d
Treatment	32	36	2	0
Control	68	49	7	0
Total	100	85	9	0

QUESTION 17 (EJOV)

Group\Choice	a	b	c	d
Treatment	60	2	8	0
Control	107	8	7	2
Total	167	10	15	2

QUESTION 18 (ECOB)

Group\Choice	a	b	c	d
Treatment	35	9	19	7
Control	47	19	28	30
Total	82	28	47	37

QUESTION 19 (EFORT)

Group\Choice	a	b	c	d
Treatment	7	19	25	19
Control	25	15	51	33
Total	32	34	76	52

QUESTION 20 (EPASC)

Group\Choice	a	b	c	d
Treatment	26	20	21	3
Control	62	22	35	5
Total	88	42	56	8

QUESTION 21 (EALG)

Group\Choice	a	b	c	d
Treatment	63	2	2	3
Control	116	1	1	6
Total	179	3	3	9

QUESTION 22 (EMOD2)

Group\Choice	a	b	c	d
Treatment	67	3	0	0
Control	118	3	3	0
Total	185	6	3	0

QUESTION 23 (EC)

Group\Choice	a	b	c	d
Treatment	46	20	4	0
Control	90	26	8	0
Total	136	46	12	0

QUESTION 24 (PGEONE)

Group\Choice	a	b	c
Treatment	11	53	6
Control	27	84	13
Total	38	137	19

QUESTION 25 (PGEPER)

Group\Choice	a	b	c
Treatment	51	14	5
Control	98	18	8
Total	149	32	13

QUESTION 26 (PGENO)

Group\Choice	a	b	c
Treatment	2	64	3
Control	6	112	6
Total	8	176	9

QUESTION 27 (PDEONE)

Group\Choice	a	b	c
Treatment	2	62	6
Control	11	104	9
Total	13	166	15

QUESTION 28 (PDEAN)

Group\Choice	a	b	c
Treatment	62	7	1
Control	99	16	9
Total	161	23	10

QUESTION 29 (PDEPER)

Group\Choice	a	b	c
Treatment	51	12	7
Control	77	33	14
Total	128	45	21

QUESTION 30 (PDENO)

Group\Choice	a	b	c
Treatment	1	65	3
Control	6	109	9
Total	7	174	12

QUESTION 31 (PFUNDS)

Group\Choice	a	b	c	d	e	f	g
Treatment	8	4	18	11	10	5	13
Control	16	11	22	34	7	20	13
Total	24	15	40	45	17	25	26

QUESTION 32 (PEDOMIAN)

Group\Choice	a	b	c	d
Treatment	6	11	52	0
Control	14	9	98	3
Total	20	20	150	3

QUESTION 33 (PEGRADE)

Group\Choice	a	b	c	d
Treatment	24	23	21	1
Control	53	29	32	10
Total	77	52	53	11

QUESTION 34 (AJPM)

Group\Choice	a	b
Treatment	25	44
Control	38	85
Total	63	129

QUESTION 35 (AJTM)

Group\Choice	a	b
Treatment	25	45
Control	46	77
Total	71	122

QUESTION 36 (AJDEV)

Group\Choice	a	b
Treatment	27	43
Control	40	83
Total	67	126

QUESTION 37 (AJACQ)

Group\Choice	a	b
Treatment	29	41
Control	37	85
Total	66	126

QUESTION 38 (AJMX)

Group\Choice	a	b
Treatment	20	50
Control	35	88
Total	55	138

QUESTION 39 (AJOTHER)

Group\Choice	a	b
Treatment	4	56
Control	20	89
Total	24	145

QUESTION 40 (AREASON)

Group\Choice	a	b	c	d
Treatment	64	0	0	5
Control	104	1	0	19
Total	168	1	0	24

QUESTION 41 (ABENEFIT)

Group\Choice	a	b	c	d
Treatment	2	0	56	12
Control	8	2	75	37
Total	10	2	131	49

QUESTION 42 (AABSTR)

Group\Choice	a	b	c
Treatment	57	6	7
Control	73	5	46
Total	130	11	53

QUESTION 43 (AIH)

Group\Choice	a	b	c
Treatment	62	4	4
Control	72	9	43
Total	134	13	47

QUESTION 44 (AMOD)

Group\Choice	a	b	c
Treatment	63	3	4
Control	95	1	28
Total	158	4	32

QUESTION 45 (ALOCAL)

Group\Choice	a	b	c
Treatment	42	5	23
Control	66	5	53
Total	108	10	76

QUESTION 46 (AUNIF)

Group\Choice	a	b	c
Treatment	48	8	14
Control	67	6	51
Total	115	14	65

QUESTION 47 (ACOMP)

Group\Choice	a	b	c
Treatment	31	13	26
Control	51	10	62
Total	82	23	88

QUESTION 48 (ACONF)

Group\Choice	a	b	c
Treatment	29	11	30
Control	38	12	73
Total	67	23	103

QUESTION 49 (SFA)

Group\Choice	a	b	c	d	e
Treatment	27	22	6	6	9
Control	33	31	17	16	27
Total	60	53	23	22	36

QUESTION 50 (SFSE)

Group\Choice	a	b	c	d	e
Treatment	51	11	4	2	2
Control	61	39	9	7	8
Total	112	50	13	9	10

QUESTION 51 (SDAPP)

Group\Choice	a	b	c	d	e
Treatment	39	18	10	1	2
Control	39	46	26	8	4
Total	78	64	36	9	6

The optical reader that read the answer forms converted the alphabetic responses, to the survey questions, to numeric responses. The conversion is as follows:

a = 0	e = 4
b = 1	f = 5
c = 2	g = 6
d = 3	h = 7

Below is the SAS printout of the descriptive statistics for each survey question.

The SAS System 14:20 Sunday, August 25, 1991

Variable	N	Mean	Std Dev	Minimum	Maximum
GRADE	194	2.2731959	2.0467599	0	6.0000000
CMND	194	2.2989691	1.8557568	0	7.0000000
AFSC	190	3.7736842	1.4090945	0	7.0000000
JYEARS	194	0.9896907	0.8637171	0	3.0000000
JPM	194	1.0773196	1.0175665	0	4.0000000
JTM	194	1.3556701	1.0736089	0	4.0000000
JDEV	193	1.1865285	1.1348756	0	4.0000000
JACQ	193	0.9533679	1.0863267	0	4.0000000
JMX	194	1.2938144	1.1477678	0	4.0000000
JOTHER	167	0.6167665	1.0570277	0	4.0000000
ECP	194	1.5103093	0.7565924	0	3.0000000
ESE	194	0.8659794	0.5411342	0	3.0000000
ESREQ	194	0.9123711	0.6960671	0	3.0000000
ESMX	194	0.8969072	0.7750586	0	3.0000000
ESDES	194	0.9845361	0.7784454	0	3.0000000
EADA	194	0.5309278	0.5861724	0	2.0000000
EJOV	194	0.2371134	0.6320162	0	3.0000000
ECOB	194	1.2010309	1.1809362	0	3.0000000
EFORT	194	1.7628866	1.0258139	0	3.0000000
EPASC	194	0.9175258	0.9513569	0	3.0000000
EALG	194	0.1855670	0.6802910	0	3.0000000
EMOD2	194	0.0618557	0.2990289	0	2.0000000
EC	194	0.3608247	0.5968061	0	2.0000000
PGEONE	194	0.9020619	0.5345046	0	2.0000000
PGEPER	194	0.2989691	0.5876968	0	2.0000000
PGENO	193	1.0051813	0.2975142	0	2.0000000
PDEONE	194	1.0103093	0.3807504	0	2.0000000
PDEAN	194	0.2216495	0.5263476	0	2.0000000
PDEP	194	0.4484536	0.6828185	0	2.0000000
PDENO	193	1.0259067	0.3135023	0	2.0000000
PFUNDS	192	3.0156250	1.8686629	0	6.0000000
PEDOMAIN	193	1.7046632	0.6699106	0	3.0000000

Variable	N	Mean	Std Dev	Minimum	Maximum
PEGRADE	193	0.9896373	0.9519149	0	3.0000000
AJPM	193	0.6839378	0.4985406	0	3.0000000
AJTM	193	0.6321244	0.4834815	0	1.0000000
AJDEV	193	0.6528497	0.4773022	0	1.0000000
AJACQ	192	0.6562500	0.4762006	0	1.0000000
AJMX	193	0.7150259	0.4525761	0	1.0000000
AJOTHER	170	0.8705882	0.3857901	0	3.0000000
AREASON	193	0.3782383	0.9931762	0	3.0000000
ABENEFIT	193	2.1502591	0.6870926	0	4.0000000
AABSTR	194	0.6030928	0.8887266	0	2.0000000
AIH	194	0.5515464	0.8577128	0	2.0000000
AMOD	194	0.3505155	0.7486255	0	2.0000000
ALOCAL	194	0.8350515	0.9622988	0	2.0000000
AUNIF	194	0.7422680	0.9305227	0	2.0000000
ACOMP	193	1.0310881	0.9404494	0	2.0000000
ACONF	193	1.1865285	0.9221944	0	2.0000000
SFA	194	1.5927835	1.4871294	0	4.0000000
SFSE	194	0.7371134	1.1095272	0	4.0000000
SDAPP	193	0.9689119	1.0303051	0	4.0000000

Bibliography

Ada Joint Program Office. *DOD Ada Software Engineering Education and Training*. Defense Technical Information Center, AD-A200 766, October 1987.

Air Force Broad Area Review (AFBAR). *Software Management Broad Area Review (BAR) Report*. Washington: HQ USAF Systems Command, 19 December 1989.

American Heritage Dictionary, The. Boston: Houghton Miffin Company, 1979.

Anderson, Teresa L. and Catherine W. McDonald. *Ada Software Engineering Education and Training Requirements Within the U.S. Army*. Paper submitted to the Institute for Defense Analyses, Alexandria VA, December 1988 (AD-A224 224).

Booch, Grady. *Object Oriented Design With Applications*. California: The Benjamin/Cummings Publishing Company, Inc., 1991.

Booch, Grady. *Software Engineering With Ada*. California: The Benjamin/Cummings Publishing Company, Inc., 1983.

Bralick, Capt William A. Jr. and Capt David A. Umphress. "Ada and the Air Force Institute of Technology," *Ada Software Engineering Education & Training Symposium*. 187-203, June 1988.

Croak, Maj Thomas. Telephone interview. HQAF Software Management office (SCXS), Andrews Air Force Base MD, 18 March 1991.

Davis, Alan M. *Software Requirements Analysis & Specification*. New Jersey: Prentice Hall, 1990.

Department of Defense. *Computer Programming Language Policy*. DOD Directive 3405.1, 2 April 1987.

Department of Defense. *Use of Ada in Weapon Systems*. DOD Directive 3405.2, 20 Mar 1987.

Deputy Assistant Secretary (Communications, Computers, & Logistics). Department of the Air Force, Washington DC. Action Memorandum for Air Force Policy on Programming Languages, 7 August 1990.

Devore, Jay L. *Probability and Statistics for Engineering and the Sciences (Second Edition)*. California: Brooks/Cole Publishing Company, 1987.

Emory, C. William. *Business Research Methods (Third Edition)*. Illinois: Irwin, 1985.

Guido, Capt Dawn A. Personal interview. Air Force Institute of Technology, Department of Electrical and Computer Engineering, Wright-Patterson Air Force Base OH, 31 January 1991.

Howatt, Maj J.W., and Maj D.A. Umphress. "Implementing AFTOMS, An Ada Perspective," December 1989.

Institute for Defense Analyses. *Ada Software Engineering Education and Training Requirements Within the U.S. Army*. Defense Technical Information Center, AD-A224 224, December 1988.

International Society of Electrical and Electronics Engineers. *ANSIIEEE Standard 729-1983 Standard Glossary of Software Engineering Terminology*. IEEE, New York, 1984.

Lawlis, Lt Col Patricia K. Personal Correspondence. Air Force Institute of Technology, Department of Electrical and Computer Engineering, Wright-Patterson Air Force Base OH, 3 September 1991.

Marsh, Alton. "Pentagon Up Against a Software Wall." *Government Executive*, 62-63 (May 1990).

McManus, James I. and G. Gordon Schulmeyer. *Handbook of Software Quality Assurance*. New York: Van Nostrand Reinhold, 1987.

Meyburg, Arnim H. and Peter R. Stopher. *Survey Sampling and Multivariate Analysis for Social Scientists and Engineers*. Massachusetts: D.C. Heath and Company, 1979.

Sommerville, Ian. *Software Engineering (Third Edition)*. New York: Addison-Wesley Publishing Company, 1989.

Tomayko, James E. "Lessons Learned Teaching Ada in the Context of Software Engineering," *The Journal of Systems and Software*, 10: 281-283 (1989).

"U.S. Ada Products Market Approaches $1 Billion in 1989," *Ada Strategies*, 4: 1-3 (April 1990).

Vita

Captain James W. Worley, Jr. was born on 27 October 1954 in Washington, D.C. He graduated from Aurora Senior High School in Aurora, Indiana in 1972. He enlisted in the USAF in January 1973 and entered active duty in March 1973. He completed tours at Offutt AFB, Nebraska, Ghedi AB, Italy, and England AFB, Louisiana. After spending nine and one-half years enlisted service as a Corrosion Control Specialist and a Command and Control Specialist he entered the Airman Education and Commissioning Program. In August 1985 he graduated from the University of Kentucky in Lexington, Kentucky with a Bachelor of Science in Electrical Engineering and immediately entered Officer Training School (OTS). After graduating OTS in November 1985, he served his first tour as a commissioned officer at the Weapons Lab, Kirtland AFB, New Mexico. There he performed a broad spectrum of engineering and management tasks in non-conventional space imaging technologies until entering the School of Systems and Logistics, Air Force Institute of Technology, in May 1990. During the Weapons Lab tour he received a regular commission in the USAF and completed SOS in residence.

Permanent Address: 202 Springdale Road

Aurora, IN 47001

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1991	Master's Thesis

4. TITLE AND SUBTITLE

EDUCATION NECESSARY FOR AIR FORCE SOFTWARE MANAGERS TO USE THE ADA PROGRAMMING LANGUAGE AND SOFTWARE ENGINEERING EFFECTIVELY

5. FUNDING NUMBERS

6. AUTHOR(S)

James W. Worley, Jr., Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GSS/ENG/91D-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Major Thomas Croak
HQAF/SCXS
Pentagon
Washington DC 20330-5190

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)* This research investigated the perceived Ada and software engineering education needs of software managers throughout the Air Force and provided recommendations to Air Staff for developing the education. Since the Department of Defense (DOD) mandated all DOD agencies use the Ada programming language for software development, the education of the personnel who have to support the policy has lagged behind the implementation. Ada is not a simple language to use and is only fully effective if used with sound software engineering principles. A survey was given to Air Force mid-level personnel who manage software in some capacity. Of the software personnel surveyed, 48 percent indicated that Ada education would enhance their ability to perform their jobs and 83 percent indicated software engineering education would help. Less than 45 percent of the personnel who work with Ada have had a formal course in using the language while over 70 percent of the people have had some form of formal software engineering education. This research also investigated the required frequency of education. The respondents indicated that the frequency (periodic or as-needed) should depend on the nature of the individuals job and rank or position.

14. SUBJECT TERMS

Ada, Software Engineering, Programming, Embedded Systems, Lifecycle

15. NUMBER OF PAGES 86

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT Control Number AFIT/GSS/ENG/91D-12

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

   a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

   a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

   Man Years ________________ $________________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

   a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments

________________________ ________________________
Name and Grade Organization

________________________ ________________________
Position or Title Address